

Narration	Time
'Ruby' میں 'Object Oriented Programming - Methods' کے ٹیوٹوریل میں خوش آمدید۔	00:01
اس ٹیوٹوریل میں آپ مندرجہ ذیل کرنا سیکھیں گے:	00:07
* instance methods	00:09
* class methods	00:11
* accessor methods.	00:14
یہاں ہم Ubuntu ورژن 12.04 اور	00:15
'Ruby' 1.9.3 استعمال کر رہے ہیں	00:19
اس ٹیوٹوریل کو سمجھنے کے لئے آپ کے پاس ایک فعال Internet کنکشن ہونا چاہئے۔	00:22
آپ کو Linux commands، Terminal اور Text-editor کی بھی معلومات ہونی چاہئے۔	00:27
اگر نہیں، تو متعلقہ ٹیوٹوریل کے لئے، ہماری ویب سائٹ دیکھیں۔	00:31
شروع کرنے سے پہلے، یاد کریں کہ ہم نے 'ttt' ڈائریکٹری پہلے بنائی تھی۔	00:36
اب اس ڈائریکٹری پر جاتے ہیں۔	00:41
پھر 'ruby-tutorial' پر	00:44
'oop-methods' نامی ڈائریکٹری بنائیں اور cd استعمال کر کے اس میں جائیں۔	00:47
Instance methods کیا ہیں؟	00:54
Instance methods وہ methods ہیں جو اس class کے سارے instances کے لئے دستیاب ہوتے ہیں۔	00:56
پہلے ہم نے سیکھا تھا کہ ایک class کے objects یا instances کس طرح بناتے ہیں۔	01:03
بنیادی سطح کے 'Ruby' ٹیوٹوریل میں ظاہر طریقے سے 'gedit' میں ایک فائل بنائیں۔	01:09
اسے 'instance_methods.rb' نام دیں۔	01:14
میرے پاس instance methods کے نفاذ کی ایک فعال مثال ہے۔	01:19
آپ ٹیوٹوریل پورا کرتے وقت، اسے درمیان میں روک کر کوڈ ٹائپ کر سکتے ہیں	01:24
میں نے اس مثال میں 'Product' نامی ایک class ڈفائن کیا ہے۔	01:29

01:33	میں نے instance variables ، 'name' اور 'price' کو انشلا نر کرنے کے لئے ایک initialize میتھڈ کال کیا ہے۔
01:41	میں نے 'name' اور 'price' نامی instance methods کو بھی ڈفائن کیا ہے۔
01:47	ان میں سے ہر ایک instance variable بالترتیب 'name' اور 'price' رٹرن کرتے ہیں۔
01:54	Instance methods عام methods کی طرح ہی ڈفائن ہوتے ہیں۔
01:58	پہلے ہم نے سیکھا تھا کہ 'Ruby' میں methods کس طرح بناتے ہیں۔
02:02	جلد ہی، ہم دیکھیں گے کہ یہ methods سارے instances کے لئے کس طرح دستیاب ہوں گے۔
02:07	اب جو logic ہمارے پاس ہے، اسے استعمال کرتے ہیں۔
02:11	یہاں، میں نے ایک 'Product' آبجیکٹ کو انشلا نر کیا ہے اور اسے 'product_object_1' کا نام دیا ہے۔
02:18	میں نے اسے 'name value' اور 'price value' کے ساتھ انشلا نر کیا ہے۔
02:24	initializer block ، instance variables ، 'name' اور 'price' کو ویلیوز پاس کرتا ہے۔
02:31	اب یہ product instance یا object ، instance variables ، 'name' اور 'price' کو استعمال کر سکتا ہے۔
02:37	ان methods کو کال کرنے پر، ہمیں instance variables میں جمع ویلیوز حاصل ہوتی ہیں۔
02:43	اب اس کوڈ کو ایکریوٹ کرتے ہیں۔
02:46	terminal پر جائیں اور ٹائپ کریں: 'ruby instance_methods.rb' اور آؤٹ پٹ دیکھنے کے لئے Enter دبائیں۔
02:56	آپ دیکھیں گے کہ آپ نے جن ویلیوز سے object کو انشلا نر کیا یہ ان کو پرنٹ کرے گا۔
03:02	یعنی 'laptop' اور "35,000"
03:07	اب ایک اور instance یا object کو انشلا نر کریں۔
03:12	اب اس object کو 'product_object_2' نام دیتے ہیں۔
03:18	اس بار، 'name' اور 'price' کو ویلیوز کا ایک دوسرا سیٹ دیتے ہیں۔
03:23	اب اس object کے لئے instance methods ، 'name' اور 'price' کال کرتے ہیں۔
03:35	آگے terminal پر واپس جاتے ہیں اور پہلے کی طرح کوڈ ایکریوٹ کرتے ہیں۔

03:41	آپ دیکھیں گے کہ یہ کامیابی سے ایکز کیوٹ ہوتا ہے اور نئی ویلیوز پرنٹ کرتا ہے۔
03:48	یہ ثابت کرتا ہے کہ instance methods ، اس class Product کے سارے objects کے لئے دستیاب ہیں۔
03:55	اب آپ اپنے خود کے instance methods لکھنے کے قابل ہونے چاہیے۔
03:59	آگے، دیکھتے ہیں کہ class methods کیا ہوتے ہیں
04:04	class methods صرف class کے لئے دستیاب methods ہوتے ہیں۔
04:09	یہ methods ، class کے instances کے لئے دستیاب نہیں ہوتے ہیں۔
04:14	آپ مختلف طریقے سے class methods کو ڈفائن کر سکتے ہیں۔
04:16	اب ایک مثال دیکھتے ہیں۔
04:18	بنیادی سطح کے 'Ruby' ٹیوٹوریلز میں ظاہر طریقے سے 'gedit' میں ایک نئی فائل بنائیں۔
04:24	اسے 'class_methods.rb' نام دیں۔
04:28	میرے پاس class methods کی ایک فعال مثال ہے۔
04:32	آپ ٹیوٹوریل پورا کرتے وقت، اسے درمیان میں روک کر کوڈ ٹائپ کر سکتے ہیں
04:36	میں نے پہلے کی طرح ایک 'Product class' ڈفائن کیا ہے۔
04:40	اور پہلے کی طرح ہی ایک 'initializer' بھی کال کیا ہے۔
04:44	تاہم، اس وقت میں نے 'description' نامی ایک اضافی 'argument' شامل کیا ہے۔
04:48	میں پہلے کے 'instance variables' کے برعکس، ویلیوز کو رکھنے کے لئے class variables استعمال کر رہا ہوں۔
04:55	یہ class ، class methods کو ڈفائن کرنے کے تین مختلف طریقے دکھائے گا
05:01	'name' کے لئے class method ڈیکلریشن کو جانچیں۔
05:06	یہاں یہ "Product" class name استعمال کرتے ہوئے ڈفائن ہوتی ہے۔
05:10	پھر، دوسرے class method ڈیکلریشن کو جانچیں۔
05:14	یہاں میں نے کی وارڈ self استعمال کیا ہے۔
05:18	آگے، تیسرا طریقہ جانچیں جس سے آپ class methods ڈفائن کر سکیں۔

05:23	اب اس class method کو لاگو کرتے ہیں۔
05:27	اب پہلے کی طرح 'Product' کا object انشلا بنز کرتے ہیں۔
05:32	اس وقت ہم 'description' کے لئے بھی ایک ویلیو دے رہے ہیں۔
05:37	اب یہاں ظاہر class methods کو کال کرتے ہیں۔
05:42	اب کوڈ کو ایک کیوٹ کرتے ہیں اور آؤٹ پٹ جانچتے ہیں۔
05:47	terminal پر جائیں اور پہلے کی طرح کوڈ ایک کیوٹ کریں۔
05:54	آپ دیکھیں گے کہ یہ 'name'، 'price' اور 'description' کے لئے ویلیوز پرنٹ کرے گا۔
05:59	اب آپ اپنے 'class methods' لکھنے کے قابل ہونے چاہیے۔
06:03	اب ہم دیکھیں گے کہ 'accessor methods' کیا ہوتے ہیں
06:07	'Ruby' classes میں ڈفائن کئے گئے ڈیٹا تک رسائی کے لئے 'accessor methods' استعمال کرتا ہے۔
06:13	'Accessor methods'، 'setter methods' اور 'getter methods' سے بنتا ہے۔
06:18	'Setter method' ویلیوز مقرر کرتا ہے۔
06:22	'Getter method' ان ویلیوز کو حاصل کرتا ہے۔
06:24	'Ruby' ان methods کو ڈیکلئیر کرنے کے لئے لفظ 'attr_accessor' استعمال کرتا ہے۔
06:31	اب 'accessor methods' کی ایک مثال دیکھتے ہیں۔
06:35	بنیادی سطح کے 'Ruby' ٹیوٹوریلز میں ظاہر طریقے سے 'gedit' میں ایک نئی فائل بنائیں۔
06:39	اسے 'accessor_methods.rb' نام دیں۔
06:43	میرے پاس 'accessor methods' کے نفاذ کی ایک فعال مثال ہے۔
06:47	آپ ٹیوٹوریل پورا کرتے وقت، اسے درمیان میں روک کر کوڈ ٹائپ کر سکتے ہیں
06:52	اس مثال میں، میں نے 'Product' نامی ایک class ڈفائن کیا ہے۔
06:56	میں نے 'name' اور 'price' کے لئے 'attr_accessor' ڈیکلئیر کیا ہے۔
07:01	ان methods کو استعمال کرنے کے لئے صرف اتنا ہی کرنے کی ضرورت ہے۔
07:05	اب اسے لاگو کرتے ہیں۔

07:07	میں نے ایک Product object انشلاز کیا ہے۔
07:10	پھر، میں نے Product object کے 'name' اور 'price' مقرر کئے ہیں۔
07:14	یہ اس لئے ممکن ہے کیونکہ ڈیفالٹ طور پر 'attr_declaration' ویلیوز قائم کرنے کے لئے methods بناتا ہے۔
07:22	پھر میں نے 'name' اور 'price' کے لئے 'getter method' استعمال کرتے ہوئے ویلیوز پرنٹ کرنے کی کوشش کی ہے۔
07:28	ان getter methods کو بھی 'attr_accessor' کے ڈکلیئریشن سے بنایا گیا تھا۔
07:35	اب پہلے کی طرح کوڈ ایکریوٹ کرتے ہیں۔
07:40	آپ دیکھیں گے کہ جو ویلیوز سیٹ کی گئی تھیں یہ ان کو پرنٹ کرتا ہے۔
07:44	اب سے، آپ اپنے 'accessor methods' لکھنے کے قابل ہونے چاہیے۔
07:50	ایک چیز جو توجہ دینے کی ہے کہ ڈفالٹ طور پر 'accessor methods'، 'instance methods' ہوتے ہیں۔
07:55	اس لئے یہ class Product کے مختلف instances سے قابل رسائی ہوتے ہیں۔
08:00	اس ٹیوٹوریل میں ہم نے سیکھا : class methods, instance methods اور accessor methods .
08:06	ایک مشق: 'Temperature' نامی ایک class ڈفائن کریں۔
08:10	Ruby کا accessor method سنٹیکس استعمال کرتے ہوئے ایک instance method لکھیں۔
08:15	یہ method دیے ہوئے 'Fahrenheit' کے لئے 'Celsius' کا حساب کرنا چاہئے۔
08:20	مندرجہ ذیل لنک پر دستیاب ویڈیو دیکھیں۔
08:23	یہ اسپوکن ٹیوٹوریل پروجیکٹ کا خلاصہ کرتا ہے۔
08:26	اچھی بینڈوڈتھ نہ ملنے پر آپ اسے ڈاؤن لوڈ کر کے دیکھ سکتے ہیں۔
08:30	اسپوکن ٹیوٹوریل پروجیکٹ ٹیم:
08:31	اسپوکن ٹیوٹوریلز کا استعمال کرتے ہوئے ورکشاپ چلاتی ہے۔
08:34	اور آن لائن ٹیسٹ پاس کرنے والوں کو سرٹیفکیٹ دیتے ہیں۔

مزید معلومات کے لئے، contact at spoken hyphen tutorial dot org پر لکھیں۔	08:38
اسپوکن ٹیوٹوریل پروجیکٹ ٹاک ٹوائے ٹیچر پراجیکٹ کا حصہ ہے۔	08:44
یہ بھارتی حکومت کے ایم ایچ آر ڈی کے آئی سی ٹی کے ذریعے قومی خواندگی مشن کی طرف سے حمایت شدہ ہے۔	08:48
اس مشن پر مزید معلومات spoken hyphen tutorial dot org slash NMEICT hyphen Intro پر دستیاب ہیں۔	08:55
اس اسکرپٹ کا ترجمہ اور صدا بندی میں نے یعنی وجاحت احمد نے کی ہے۔ شامل ہونے کیلئے آپ کا شکریہ۔	09:03